AMENDMENTS TO THE SPECIFICATION:

Please replace paragraph [0022] with the following amended paragraph:

[0022] Service providers are increasingly moving toward dynamically generating boot files based ~~up on~~ upon the content of the device boot file request message. (A dynamic TFTP ~~(DTPTP)~~(DTFTP) server meeting the requirements of the present invention is described in commonly owned U.S. Patent ~~Application~~ 7,293,078, entitled "System And Method For Provisioning A Provisionable Network Device With A Dynamically Generated Boot File Using A Server" and filed ~~Jun. XX, 2003~~July 14, 2003, which ~~. That~~ application is incorporated herein in its entirety and for all purposes.) For example, a DTFTP server may parse the file name of the device boot file in the device boot file message to determine what configuration attributes are required in the device boot file and what values are to be assigned to those attributes.

Please replace paragraph [0023] with the following amended paragraph:

[0023] While the introduction of DTFTP servers for boot file generation greatly improves the ability of service providers to manage boot file generation, use of DTFTP servers alone does not solve problems of managing changes in provisioning parameters that affect the boot file generation server and the CMTS. What is needed is ~~systema~~ a system and method for updating and synchronizing changes in provisioning parameters used by DTFTP by servers and CMTSs that does not require manual intervention.

Please replace paragraph [0054] with the following amended paragraph:

[0054] Additionally, the central datastore **400** manages the synchronization of provisioning parameter values between a DTFTP server and the CMTSs supported by that DTFTP server. Referring to **FIG. 5**, a process of synchronizing a DTFTP server and one or more CMTSs supported by that DTFTP server

according to an embodiment of the present invention is illustrated. The DTFTP is polled **500**. The poller returns provisioning parameters and data to a datastore **505**. The datastore determines whether the "current" set of provisioning parameters and/or provisioning parameter values reported by the poller differ from the set of provisioning parameter values stored in central datastore **510**. If the ~~If~~ current set of provisioning parameters and provisioning parameters values stored in the datastore ~~are~~ is the same as ~~those~~ the set returned by the poller, the process resumes with polling the DTFTP server **500**. If changes have been made, the polled set of provisioning parameters and provisioning parameter values is stored in the central datastore **515**. The central datastore determines whether any of the changes in provisioning parameters and/or the provisioning parameter values are needed by one or more ~~CMTS~~ CMTSs supported by the polled DTFTP server **520**. If the changes are needed by one or more ~~CMTS~~ CMTSs supported by the polled DTFTP server, the changes are passed on to the appropriate CMTSs by the central datastore **525** and those CMTSs are reconfigured **530**. If the changes are needed by a CMTS supported by the polled DTFTP server, the process resumes with polling the DTFTP server **500**.

Please replace paragraph [0055] with the following amended paragraph:

[0055] In another exemplary embodiment, ~~the~~ a DTFTP server and one or more CMTSs supported by that DTFTP server use a shared secret to compute a media integrity check (MIC) value for a boot file. In this exemplary embodiment, the value of the shared secret is changed at the DTFTP server. The current provisioning parameters (including the new shared secret value) are reported by a poller to ~~central~~a central datastore. Because a provisioning parameter value has changed, the polled set of provisioning parameter values is stored at the central datastore. The central datasore determines that the new shared secret value is needed by the CMTSs supported by the DTFTP server and sends the new shared secret value to those CMTSs via an SNMP message. If a CMTS supports storing two shared secrets, the new shared secret is stored at the CMTS but is not "active"

in that the current shared secret is still used to generate the MIC value. Following delivery of the new shared secret to all CMTSs supported by the DTFTP server, the central datastore sends a second message instructing each CMTS to delete the current shared secret and to activate the "new" shared secret. The result of this process is to synchronize the shared secret used by the DTFTP server and each of the CMTSs supported by the DTFTP server.

Please replace paragraph [0056] with the following amended paragraph:

[0056] In still another embodiment, a central datastore holds the current configuration of all DTFTP servers. Referring to **FIG. 6**, in addition to receiving changes made to the provisioning parameter values ~~made~~ by ~~at~~ a DTFTP server, the central datastore **400** is linked to DTFTP server A **402** and to DTFTP server B **410**. Through this link, the central datastore **400** can invoke a global change in a provisioning parameter value across DTFTP server A **402** and DTFTP server B **410** and the CMTSs supported by those DTFTP servers. In this embodiment, a global change is broadcast to all of the DTFTPs within an HFN. The ~~changed~~ change is then reported back to the central datastore **400** during the polling process. The central datastore **400** then propagates the changes to all the CMTSs within the HFN.